

目录

目录	1
频道事件	2
一、整体说明	2
1. 回调请求方式	2
2. 回调消息示例	2
二、请求字段定义说明	2
1. 请求Header	2
2. 请求Body	2
Event Type详情	2

频道事件

一、整体说明

1. 回调请求方式

消息回调信息通过HTTP的POST方式，请求客户的服务器，数据格式为JSON.

2. 回调消息示例

用curl模拟请求的示例如下：

```
curl -H "Content-Type:application/json"
      -H "KRTC-Signature:xxxxxxx" //客户服务器验证签名时用，如果暂不验证签名，可忽略此字段
      -X POST
      --data {"noticeId":"1631776189707676964","productId":1,"eventType":101,"notifyMs":1631776189707,"payload":{"channelName":"default","ts":1631776189}} //回调给客户数据
      http://1.1.1.1:8080/cbUri // 客户的回调服务器
```

二、请求字段定义说明

1. 请求Header

回调请求的Header如下		请求字段	字段值
	:		
Content-Type		application/json	
KRTC-Signature		客户服务器验证签名时用，如果暂不验证签名，可忽略此字段。如需验证，可看以下签名验证部分	

2. 请求Body

POST请求示例：

```
{
  "noticeId": "1631776189707676964",
  "productId": 1,
  "eventType": 101,
  "notifyMs": 1631776189707,
  "payload": {...} // payload 内容 不同的eventType，不同的格式。
}
```

请求Body字段说明：		字段	类型	说明
noticeId	string	通知Id，标识一次事件的通知。值为：调用时的unix时间戳字符串，精确到纳秒		
productId	int	业务Id，1表示实时通信		
eventType	int	通知事件类型，具体定义，详见【事件类型定义】		
notifyMs	int	调用客户服务器回调时的unix时间戳，精确到毫秒		
payload	JSON格式	通知事件的具体内容，与eventType相对应		
eventType事件类型定义如下：		Event Type	描述	
101			创建房间	
102			销毁房间，最后一个用户离开房间时	
103			直播场景，主播加入房间	
104			直播场景，主播离开房间	
105			直播场景，观众加入房间	
106			直播场景，观众离开房间	
111			直播场景，用户角色切换为主播	
112			直播场景，角色切换为观众	

Event Type详情

1. 创建房间事件，Event Type 101。payload格式如下：

```
{
  "channelName": "default",
```

```
    "ts": 1631776189
  }
```

字段	类型	说明
channelName	string	房间名称
ts	int	房间创建的事件，unix时间戳

2. 销毁房间事件，Event Type 102。payload格式如下：

```
{
  "channelName": "defau",
  "ts": 1631776189
}
```

字段	类型	说明
channelName	string	房间名称
ts	int	房间创建的事件，unix时间戳

3. 直播场景，主播加入房间，Event Type 103。payload格式如下：

```
{ "channelName": "defau", "uid": "123", "platform": 1, "clientSeq": 0, "ts": 1631776195 }
```

字段	类型	说明
channelName	string	房间名称
uid	string	用户ID
platform	int	主播所属平台：1 Android；2：IOS
clientSeq	int	暂未支持
ts	int	主播加入房间时的时间，unix时间戳

4. 直播场景，主播离开房间，Event Type 104。payload格式如下：

```
{ "channelName": "defau", "uid": "123", "platform": 1, "clientSeq": 0, "reason": 420, "ts": 1631776195,
  "duration": 10 }
```

字段	类型	说明
channelName	string	房间名称
uid	string	用户ID
platform	int	主播所属平台：1 Android；2：IOS
clientSeq	int	暂未支持
reason	int	离开房间的原因：420：正常离开；421：异常离开；469：用户被禁用离开
ts	int	主播加入房间时的时间，unix时间戳
duration	int	主播在频道时长

5. 直播场景，观众加入房间，Event Type 105。payload格式如下：

```
{ "channelName": "defau", "uid": "124", "platform": 1, "clientSeq": 0, "ts": 1631776195 }
```

字段	类型	说明
channelName	string	房间名称
uid	string	用户ID
platform	int	主播所属平台：1 Android；2：IOS
clientSeq	int	0
ts	int	主播加入房间时的时间，unix时间戳

6. 直播场景，观众离开房间，Event Type 106。payload格式如下：

```
{ "channelName": "defau", "uid": "124", "platform": 1, "clientSeq": 0, "reason": 420, "ts": 1631776195,
  "duration": 10 }
```

字段	类型	说明
channelName	string	房间名称
uid	string	用户ID
platform	int	主播所属平台：1 Android；2：IOS
clientSeq	int	暂未支持
reason	int	离开房间的原因：420：正常离开；421：异常离开；469：用户被禁用离开
ts	int	主播加入房间时的时间，unix时间戳
duration	int	主播在频道时长

7. 直播场景，用户角色切换为主播，Event Type 111。payload格式如下：

```
{ "channelName": "defau", "uid": "124", "clientSeq": 0, "ts": 1631776195, }
```

字段	类型	说明
channelName	string	房间名称
uid	string	用户ID
clientSeq	int	暂未支持

|ts|int|主播加入房间时的时间，unix时间戳|

8. 直播场景，角色切换为观众，Event Type 112。payload格式如下：

```
{ "channelName": "defau", "uid": "125", "clientSeq": 0, "ts": 1631776195, }
```

|字段|类型|说明|

|-|-|-|

|channelName|string|房间名称|

|uid|string|用户ID|

|clientSeq|int|暂未支持|

|ts|int|主播加入房间时的时间，unix时间戳|

三、简单的安全验证策略

1. 首先客户需要向KRTC运维人员申请一个秘钥串callBackSk；

2. KRTC服务请求客户服务器时，在请求的Header头中会通过“KRTC-Signature”携带一个reqSign；

3. 客户端服务器需要从“KRTC-Signature”请求头中获取 reqSign，并将请求body的字符串和callBackSk通过ComputeHmacSha256(body, callBackSk)方法生成 cpSign。具体的ComputeHmacSha256方法如下：

```
func ComputeHmacSha256(message string, secret string) string { key := []byte(secret) h := hmac.New(sha256.New, key) h.Write([]byte(message)) sha := hex.EncodeToString(h.Sum(nil)) return sha }
```

4. 如果 reqSign 与 cpSign 相同，则信任请求，解析请求的json数据；